

# Rice version 0.1.1

Jim Menard, [jimm@io.com](mailto:jimm@io.com)

March 4, 2002

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Recent Changes . . . . .	2
1.2	Conventions . . . . .	2
<b>2</b>	<b>Installing Rice</b>	<b>3</b>
<b>3</b>	<b>Starting Rice</b>	<b>3</b>
<b>4</b>	<b>Architecture</b>	<b>3</b>
4.1	The Core . . . . .	3
4.1.1	URL . . . . .	4
4.1.2	HTTP . . . . .	4
4.1.3	XML . . . . .	4
4.1.4	Timestamps . . . . .	4
4.1.5	Common Behavior . . . . .	4
4.1.6	Logging . . . . .	5
4.2	Demo-Specific Subscription Services . . . . .	5
4.3	The User Interface . . . . .	5
4.4	Rice::Syndicator . . . . .	5
4.5	Rice::Subscriber . . . . .	5
<b>5</b>	<b>Resources</b>	<b>5</b>
<b>6</b>	<b>Bugs and Limitations</b>	<b>6</b>
<b>7</b>	<b>Copying and Warranty</b>	<b>6</b>

# 1 Introduction

Rice is a Ruby implementation of the Information and Content Exchange (ICE) protocol, version 1.1. Rice implements an ICE syndicator that offers and publishes subscriptions and an ICE subscriber that subscribes to and consumes them.

Rice may be useful as a reference implementation or a public syndicator that ICE developers can use to test their subscribers.

Rice was developed based on *ICE Implementation Cookbook: Getting Started with Web Syndication* by Adam Souzis, Laird Popkin, Sami Khoury, and Bruce Hunt. This document may be found on the ICE Standard Web page at <http://www.icestandard.org>.

Ruby is an object-oriented scripting language by Yukihiro Matsumoto. The official Ruby Web site (<http://www.ruby-lang.org>) contains information and pointers to resources for this wonderful language.

Information and Content Exchange (ICE) is an XML-based protocol for content syndication and subscription. For more information, see the ICE Standard Web page.

For pointers to information about Ruby, ICE, XML and more see Section 5, *Resources*.

Rice is developed and maintained by Jim Menard ([jimm@io.com](mailto:jimm@io.com)). The latest version of Rice can be found on the official Rice Web page (<http://rice.sourceforge.net>). New releases are also announced on Freshmeat (<http://freshmeat.net>).

## 1.1 Recent Changes

Here is the summary of the bug fixes in version 0.1.1. For a complete list of changes, see the `ChangeLog` file.

- Changed “checked” to “selected” for radio buttons.
- Misspelled tag name “ise-response” changed to “ice-response”.
- The testing module is called “RubyUnit”, not “RUnit”.

## 1.2 Conventions

File and directory names are printed using a `monospace font`.

Text that will be substituted with a user-defined value is printed using *italics*.

## 2 Installing Rice

Rice itself does not need to be installed. Rice depends upon four other Ruby modules that need to be installed; one of those is optional. To install these modules,

- Retrieve NQXML, erb, and WEBrick from the Ruby Application Archive (<http://www.ruby-lang.org/en/raa.html>). Optionally retrieve RubyUnit from the same place. RUnit is only used for the scripts in the tests directory.
- Install these modules by running their `install.rb` scripts. The module erb must be installed before WEBrick. You may need to be root in order to have the proper privileges required to install them.

Finally, run the script `setup.rb`, which will create a temp directory if necessary.

## 3 Starting Rice

First start the WEBrick server. Open a command line window (a terminal window or DOS command window, depending upon your operating system).

If you want to see the server's output (page requests and error messages), start the server from the command line by running the script `httd.rb`.

Open two Web browser windows. One will be the syndicator, or publisher, and the other will be the subscriber.

If change code, may need to restart Web server and/or delete `/tmp/ice-*` files (Ruby CGI session data and Ruby pstore data).

## 4 Architecture

Rice is made up of three layers: the core, demo-specific subscription services, and the user interface.

### 4.1 The Core

At its core, any ICE implementation must speak HTTP, parse and output XML, and handle ICE data types like timestamps and URLs. In addition, it makes sense to provide behavior common to syndicators and subscribers.

Rice is no different. It uses Ruby classes to speak HTTP and retrieve URL contents. It relies upon NQXML to parse and write well-formed XML.

It provides timestamp parsing and formatting. Finally, the Rice core contains classes that can build ICE messages and that implement common syndicator and subscriber behavior.

Logging, an optional part of the ICE spec

#### 4.1.1 URL

The `Rice::URL` class represents a URL. It can parse strings to create URLs and can fetch the content of a URL based on its schema. Only HTTP and FTP fetching are implemented in Rice.

In Ruby v1.7, a `URL` class was introduced. The stable version of Ruby as of this writing is 1.6.5; version 1.7.X is a development version, though it is easily available via CVS. When Ruby v1.7 is released as a stable version and becomes widely used, `Rice::URL` may no longer be necessary.

#### 4.1.2 HTTP

Ruby's `Net::HTTP` and `Net::FTP` classes provide transport and content retrieval support.

To use HTTPS (SSL), must download additional packages from the Ruby Application Archive.

#### 4.1.3 XML

Building XML messages: `payload =i request/response =i ping/confirmation/error`

#### 4.1.4 Timestamps

I may want to change the code so `rice/datetime.rb` extends `Time` class instead of using a static methods that take a `Time` object as an argument. For example, instead of using the class method `DateTime.timeToICEDateTime` which takes a `Time` object as one of its arguments, we could provide `Time.to_ice_dt`.

#### 4.1.5 Common Behavior

Party: `syn/sub` superclass. `ICEMachine` (common ICE `syn/sub` behavior): pinging, returning responses, error responses and codes.

The `Rice::Syndicator` and `Rice::Subscriber` classes share many common behaviors that are implemented in their superclass `Rice::ICEMachine`. They are both implemented as Web-based services that listen for and receive

ICE packages via a form POST to a URL and reply with the proper ICE package.

Error codes and I18N.

#### 4.1.6 Logging

### 4.2 Demo-Specific Subscription Services

Ping, time, date, unknown, error

Inline and by reference; inline = ice-item, by reference = ice-ref

### 4.3 The User Interface

Syn. and sub. "mini-sites": Web pages using eRuby

Listener dispatches requests

Mini-sites: Interface plus role-specific behavior. Some behavior should be factored out of mini-sites.

eRuby is like PHP or JSP.

### 4.4 Rice::Syndicator

A syndicator...

### 4.5 Rice::Subscriber

A subscriber...

## 5 Resources

The official Ruby Web site (<http://www.ruby-lang.org>) contains an introduction to Ruby, the Ruby Application Archive (<http://www.ruby-lang.org/en/raa.html>), and pointers to more information.

*Programming Ruby, The Pragmatic Programmer's Guide*, by David Thomas and Andrew Hunt, is a well-written and practical introduction to Ruby. Its Web page (<http://pragmaticprogrammer.com/ruby/index.html>) as can the *ICE Implementation Cookbook: Getting Started with Web Syndication*.

*Ruby: An Introduction* is a talk I have given a few times. You can find it online as HTML and PDF on my Ruby talk page (<http://www.io.com/~jimm/downloads/rubytalk/>).

Have I mentioned the Rice home page (<http://rice.sourceforge.net>) yet? I have? Never mind.

## 6 Bugs and Limitations

The file `TODO` contains a list of known bugs and limitations. Any bugs discovered after a release are documented on the Rice Web site.

## 7 Copying and Warranty

Rice is copyrighted free software by Jim Menard and is released under the same license as Ruby. See the Ruby license at <http://www.ruby-lang.org/en/LICENSE.txt>.

Rice may be freely copied in its entirety providing this notice, all source code, all documentation, and all other files are included.

Rice is copyright © 2001-2002 by Jim Menard.

THIS SOFTWARE IS PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.